Name: Yogeshree Chandankhede

DBDA\_52

# 1) Create a user defined function which will take salary of 10 employee from the employer

# and it will give 10% increment to all the employee whose current is less than or equal to

# 10,000 rupees and 5% increment to all the employee whose having salary more than 10,000

# rupees per month. Print old salary and new salary in dataframe.

print("Enter salary for 10 emp")

increment<- function(v){

a=scan(nmax=10)

new\_vec=v

new\_vec[new\_vec<=10000]=new\_vec[new\_vec<=10000]\*1.10

new\_vec[new\_vec>10000]=new\_vec[new\_vec>10000]\*1.05

d1=data.frame(old\_sal=v,new\_sal=new\_vec)

print(d1)

}

increment(a)

# 2) Create a user define function which will take Principal, Rate of interest, Duration from the

# user and calculate Simple Interest.

SMI<- function(){

P= as.integer(readline("Enter principal"))

R= as.integer(readline("Enter Rate"))

N= as.integer(readline("Enter year"))

SI=P\*R\*N/100

SI

}

SMI()

# 3) A factory produces light bulbs and the probability that a randomly selected light bulb is

# defective is 0.1. If a quality control inspector randomly selects 15 light bulbs, what is the

# probability that –

# i) exactly two of them are defective

# ii) atleast two are defective

# iii) no bulb is defective

p=0.1

n=15

?dbinom

dbinom(2,15,0.1)

1-pbinom(1,15,0.1)

dbinom(0,15,0.1)

# 4) Two kinds of manures were used in seventeen plots of the same size other conditions being

# same. The yields in quintals are as follows

# Manure I 35 42 40 42 34 24 42

# Manure II 34 44 32 40 52 41 50 40 42 45

# Test whether the two manures differs using t test.

I=c( 35, 42, 40, 42, 34 ,24 ,42)

II=c( 34, 44, 32, 40, 52, 41, 50, 40, 42, 45)

t.test(I,II,alternative ="two.sided", paired = FALSE)

#reject

?t.test

# 5) Load tidyverse library and read diamonds, answer the following question

# i) Comment on the correlation between x & y

# ii) Fit a regression model between variable x & y

# iii) Display 5 number summary of “depth”

# iv) Plot barplot of “cut”

# v) Display all the "SI2 Clarity" whose price is less than 500

library("tidyverse")

View(diamonds)

cor(diamonds$x,diamonds$y)

lm(diamonds$y~diamonds$x)

summary(diamonds$depth)

?barplot

barplot(table(diamonds$cut),main="Barplot of Diamond Cut",xlab = "Cut",ylab = "frequency",col = "green")

?filter

subset(diamonds$clarity=="SI2", price<500)

?subset

diamonds %>% filter(clarity=="SI2"&price<500)

# 6. The calls due to the failure of a computer occur in accordance with Poisson distribution

# with a mean of 2 per day. Find the probability that

# i) there are three calls for computer failure on the next day

# ii) two or more calls on the next day

m=2

?dpois

dpois(3,2)

1-ppois(1,2)

# 7) The following are the gain in weights of cows fed on two types of diets X & Y

# Diet X 30 37 35 37 29 19 37

# Diet Y 29 39 27 35 47 37 45 35 37 40

# Test whether the two diets differs using t test.

X =c(30, 37, 35, 37, 29 ,19 ,37)

Y= c(29, 39, 27, 35, 47, 37, 45, 35, 37, 40)

t.test(X,Y,alternative="two.sided",paired=F)

#Reject

# 8) Create a data frame df with columns City, Population, and Temperature for 10 different

# cities.

# i. Group the data by City and find the average Temperature.

# ii. Find the maximum Population for each city.

# iii. Calculate the total Population across all cities

df <- data.frame(

City = c("Delhi", "Mumbai", "Kolkata", "Chennai", "Bangalore",

"Hyderabad", "Ahmedabad", "Pune", "Jaipur", "Lucknow"),

Population = c(19000000, 20000000, 15000000, 11000000, 12000000,

10000000, 8000000, 7000000, 6000000, 5000000),

Temperature = c(35.5, 33.0, 36.1, 34.2, 29.0, 32.4, 37.5, 30.1, 39.2, 38.0)

)

print(df)

avg\_temp<- df %>% group\_by(City) %>% summarise(Avg\_Tem= mean(Temperature))

print(avg\_temp)

max\_temp<- df %>% group\_by(City) %>% summarise(Max\_Tem= max(Temperature))

print(max\_temp)

total\_pop <- sum(df$Population)

total\_pop

# 9) Create a user define function which will take two numeric values as an inputs from the

# user, Ask the user to enter an operation (+, -, \*, /) they want to perform on these numbers &

# Perform the operation and print the result.

oper<- function(){

a=as.integer(readline("enter value for a"));a

b=as.integer(readline("enter value for b"));a

operation <- readline(prompt = "Enter operation (+,-,\*,/)")

result <- switch(operation,

"+" = a + b,

"-" = a - b,

"\*" = a \* b,

"/" = if (a != 0) a / b else "Cannot divide by zero",

"Invalid operation")

print(result)

}

oper()

# 10) Create a 2x3 matrix A with values from 1 to 6 and a 3x2 matrix B with values from 6 to 1

# and Perform matrix multiplication of A and B.

A= matrix(1:6,nrow = 3,ncol = 2,byrow = TRUE);A

B= matrix(6:1,nrow = 2,ncol = 3,byrow = TRUE);B

C <- A %\*% B

C

# 11) Load tidyverse library and read diamonds, answer the following question

# i) Comment on the correlation between x & z

# ii) Fit a regression model between y & z

# iii) Display 5 number summary of “price”

# iv) Plot barplot of “clarity”

# v) Draw Scatterplot & comment on the relation between depth and price

View(diamonds)

cor(diamonds$x,diamonds$y)

lm(diamonds$y~diamonds$x)

fivenum(diamonds$price)

barplot(table(diamonds$clarity),main="Barplot of Diamond of Cut", xlab = "Cut",ylab = "Frequency",col = "green")

?plot

plot(price ~ depth,type='p',data = diamonds,main="Dept vs Price",xlab = "depth",ylab = "price",pch=15)

#nhi pata puchna padenga

# 12) Use the ToothGrowth dataset.

# • Calculate the average tooth length (len)

# • Draw a boxplot of len grouped by dose.

View("ToothGrowth")

data("ToothGrowth")

head(ToothGrowth)

mean(ToothGrowth$len)

boxplot(len~dose,data = ToothGrowth,main="Barplot of ToothGrowth of Cut", xlab = "Len",ylab = "Dose",col = "green")

# 13) Write a function that takes a year as input and:

# a) Checks whether it is a leap year or not.

# b) If it is a leap year, print all months with 29 days.

# c) If not, print "Not a leap year" and list all months with 30 or more days

leap\_year <- function(){

yr=as.integer(readline("enter year"));yr

if((yr %%4==0 && 100!=0) ||yr %%4 && 400==0){

print(paste(yr,"is leap year"))

print("Month with 29 day: February")

}else{

print(paste(yr,"is not leap year"))

print("all months with 30 or more days: Jan,march,april,may,june,july,aug,sep,oct,nov,dec")

}

}

leap\_year()

# 14) Write a script that accepts a number from the user and:

# a) Checks if the number is a palindrome (same forward and backward).

# b) If not, reverse the number and display it.

# c) Also display the sum of its digits.

# 15) Create a list of 10 random numbers from 1 to 100.

# a) Write a function using loops and if-else to categorize numbers as “Even” or “Odd”.

# b) Use for loop and next statement to print only even numbers.

# c) Use repeat loop to compute the sum of numbers until it exceeds 300

set.seed(121)

mylist= sample(1:100,10)

print(mylist)

even\_odd <- function(number){

for (n in number) {

if(n%%2==0){

print(paste(n,"is Even number"))

}else if(n==0){

print(paste(n,"is zero"))

}else{

print(paste(n,"is Odd Number"))

}

}

}

even\_odd(mylist)

for (n in mylist) {

if(n%%2!=0){

next

}

print(paste(n,"even number"))

}

m=0

repeat{

n=sample(1:100,10)

m=m+n

numbers\_added <- c(numbers\_added, num)

if(m==300){

break

}#-------------- not working-------------------

}

print(m)

# 16) Using mtcars:

# a) Create a bar chart of gear counts

# b) Create a boxplot of mpg by cylinder

# c) Customize title and axis labels of the boxplot

View(mtcars)

?barplot

barplot(table(mtcars$gear),col = "lightgreen",main = "count Grear",xlab = "gear",ylab = "Range")

boxplot(mpg~cyl,data = mtcars,main="MPG VS CYL",xlab = "CYL",ylab = "MPG",col=c("lightgreen","lightblue","purple"))

#17

data=matrix(c(200,40,60,120,30,50),nrow = 2,ncol=3,dimnames = list(c("Given the new drugs","Not given the drugs"),c("Cured","Condition","No effect")));data

chisq.test(data)

#18

data=matrix(c(25,20,25,2025,35,25,25,30),nrow = 3,ncol = 3,dimnames = list(c("20-35","35-50","Above 50"),c("A","B","C")));data

chisq.test(data)

# 19) Write a user-defined function that takes marks of 5 subjects and returns: Total Mark

# Percentage and Grade (A if ≥75%, B if 60–74%, C if 50–59%, D otherwise)

Score <- function(){

print("Enter marks")

marks<- scan(n = 5)

print(marks)

obt\_marks <- sum(marks)

percentage <-(obt\_marks/500)\*100;percentage

grades <- if(percentage>=75){

"A"

}else if(percentage>=60){

"B"

}else if(percentage>=50){

"C"

}else{

"D"

}

print(paste(obt\_marks,"Obtain Marks"))

print(paste(percentage,"Percentages"))

print(paste(grades,"grade"))

}

Score()

# 20) Create a user-defined function that takes a numeric vector of 12 temperatures (in Celsius),

# and returns a dataframe with converted values in Fahrenheit and a classification (Cold if<15°C, Moderate if 15–30°C, Hot if >30°C).

c\_vec=scan(nmax=12)

cal\_tem <- function(c\_vec){

f\_v <-(c\_vec\*9/5)+32

classification <- ifelse(c\_vec < 15, "Cold",

ifelse(c\_vec <= 30, "Moderate", "Hot"))

df= data.frame(celsius=c\_vec,Fahrenheit=f\_v,classification=classification)

print(df)

}

cal\_tem(c\_vec)

# 21) The probability that a machine part is defective is 0.2. A sample of 12 parts is taken.

# Find the probability that:

# i) Exactly 4 parts are defective

# ii) More than 2 parts are defective

# iii) All parts are non-defective

p=0.2

n=12

?dbinom

dbinom(4,12,0.2)

1-pbinom(2,12,0.2)

dbinom(0,12,0.2)

# 22) Create a named vector for fruit sales: Apple – 120, Banana – 150, Orange – 100.

# (b) Draw a vertical barplot with blue bars.

# (c) Add the value of each sale on top of the bars.

# (d) Change bar color for each fruit.

fruits<- c(Apple = 120, Banana = 150, Orange = 100);fruits

barplot(fruits,main = "fruit sales",xlab = "Fruits",ylab = "Sales",col = c("lightblue","lightgreen","purple"))

text(x=bars,y=fruits,labels = fruits,col="black",pos=3,cex=1)

# 23) (a) Use the built-in mtcars dataset.

# (b) Create a boxplot of mpg grouped by number of cylinders (cyl).

# (c) Add different colors for each box.

# (d) Add a title and y-axis label.

View(mtcars)

boxplot(mpg~cyl,data = mtcars,main="MPG VS CYL",xlab = "cyl",ylab = 'mpg',col=c("lightblue","lightgreen","purple"))

# 24) (a) Create a vector for votes: A – 60, B – 30, C – 10.

# (b) Draw a pie chart with percentages shown.

# (c) Assign custom colors.

# (d) Add a main title.

vec<- c(A = 60, B = 30, C = 10);vec

?pie

# 25) Use mtcars dataset.

# (a) Find mean mpg by number of cylinders.

# (b) Find total hp by number of gears.

# (c) Count number of cars by carburetor type.

mtcars %>% group\_by(cyl) %>% summarise(avg\_mpg=mean(mpg)) #using group by

aggregate(mpg~cyl, data = mtcars,FUN = mean) #direct

aggregate(hp~gear,data = mtcars,FUN = sum)

table(mtcars$carb)

# 26) Create a 4x4 matrix with values 1 to 16.

# (a) Replace diagonal elements with 0.

# (b) Multiply all border elements by 2.

# (c) Print the matrix.

m= matrix(1:16,nrow = 4,ncol = 4,byrow = TRUE);m

diag(m)<-0;m

m[1, ] <- m[1, ] \* 2

m[4, ] <- m[4, ] \* 2

m[2:3,1] <- m[2:3,1]\*2

m[2:3, 4] <- m[2:3,4] \* 2

print(m)

?diag

# 27) Create a vector of 20 random integers.

# (a) Replace all even numbers with “Even” and odd with “Odd”.

# (b) Count number of even numbers.

# (c) Replace values > 50 with “High”.

set.seed(132)

v= sample(1:100,20);v

even\_odd <- ifelse(v%%2==0, "Even","Odd")

print(even\_odd)

cnt<- sum(v%%2==0)

print(cnt)

v[v>50] <-"High";v

# 28) Create a function to accept 2 numbers and perform:

# (a) If both even – return their sum.

# (b) If both odd – return their product.

# (c) Else return "Mixed".

even\_odd <- function(){

a<- as.integer(readline("eneter value of a"))

b<- as.integer(readline("eneter value of b"))

if(a %% 2 == 0 && b %% 2 == 0){

return( a+b)

}else if(a %%2!=0 && b%%2!=0){

return(a\*b)

}else{

return("Mixed")

}

}

even\_odd()

# 29) Create a numeric vector of 30 random values from 10 to 100.

# (a) Find all values greater than 70.

# (b) Replace values divisible by 5 with NA.

# (c) Remove NA and sort the remaining vector in increasing order.

set.seed(121)

v<- sample(1:100,30)

print(v[v>70])

v[v%%5==0]<- NA;v

clean= na.omit(v)

sor\_v= sort(clean)

# 30) Create a named vector of subject scores for a student.

# (a) Access the score for "Math".

# (b) Change "English" score to 90.

# (c) Add a new subject "Science" with score 85.

v=c(Math=75,English=56,History=79);v

print(v["Math"])

v["English"]<-90;v

v["Science"]<-85;v

# 31) Create a data frame for 5 employees with name, department, and salary.

# (a) Add a bonus column = 10% of salary.

# (b) Use subset() to find employees from "HR".

# (c) Filter employees with total (salary + bonus) > 50000.

emp<- data.frame(Name= c("A","B","C","D","E"),

Dept= c("IT","HR","Com","Ele","Sales"),

Sal= c(100000,550000,650000,7511,254600));

emp$Bonus<- emp$Sal\*0.10;emp

subset(emp,Dept=="HR")

total=subset(emp,(Sal+Bonus)>50000);total

# 32) Generate 20 random numbers between 1 and 50.

# (a) Print only odd numbers.

# (b) Stop loop if a number is divisible by both 3 and 7.

# (c) Skip values less than 10 using next.

set.seed(22)

v= sample(1:50,20);v

print(v[v%%2!=0])

for(v in v){

if(v%%3==0 && v%%7==0){

break

}else{

print(v)

}

}

set.seed(22)

v= sample(1:50,20);v

for(v in v){

if(v<10){

next

}else{

print(v)

}

}